Sentiment Analysis of Tweets using Logistic Regression

Tahir Muzaffar (2021665)[1]

[1] CS351, Artificial Intelligence, Dr. Khurram Khan Jadoon

Abstract **:**

Sentiment analysis, a subfield of natural language processing, plays a vital role in understanding public opinion and sentiment expressed in textual data. In this project, we explore the application of logistic regression for sentiment analysis of tweets. We leverage a dataset containing labeled tweets and employ preprocessing techniques to clean and prepare the text data. The logistic regression model is trained using a pipeline approach, incorporating text vectorization and feature transformation. Evaluation metrics such as F1 score and confusion matrix are utilized to assess the performance of the model. Additionally, visualizations, including word clouds and ROC curves, provide insights into the characteristics of the dataset and the effectiveness of the model.

1 Introduction

The proliferation of social media platforms has led to an unprecedented volume of user-generated content, offering a rich source of information for understanding public sentiment and opinion. Sentiment analysis, a branch of natural language processing (NLP), aims to automatically extract and quantify sentiments expressed in textual data. It has diverse applications ranging from market research and brand monitoring to political analysis and customer feedback analysis.

X (formerly, Twitter) being one of the most popular social media platforms, is a treasure trove of real-time textual data reflecting diverse opinions, emotions, and sentiments. Analyzing tweets can provide valuable insights into public attitudes towards various topics, events, products, or services. However, the sheer volume and unstructured nature of X data pose significant challenges for sentiment analysis.

In this project, we delve into the task of sentiment analysis of tweets using machine learning techniques, with a particular focus on logistic regression. Logistic regression is a widely used classification algorithm known for its simplicity, interoperability, and efficiency, making it suitable for binary classification tasks like sentiment analysis. By leveraging logistic regression, we aim to develop a robust model capable of accurately classifying tweets into positive (0) and negative (1) sentiment categories.

2 Background

Sentiment analysis, also known as opinion mining, is a technique used to determine the sentiment or emotional tone behind a body of text. It is a significant aspect of natural language processing (NLP) that helps in understanding the attitudes, opinions, and emotions expressed in written language. The primary goal of sentiment analysis is to classify text into predefined sentiment categories, typically positive, negative, or neutral.

Logistic regression is a statistical method used for binary classification problems. It models the probability of a certain class or event, such as whether a tweet expresses positive or negative sentiment, based on one or more predictor variables. In the context of sentiment analysis, logistic regression can be highly effective when combined with appropriate text preprocessing and feature extraction techniques. Text preprocessing involves cleaning the text data to remove noise, such as special characters, URLs, and user mentions, which can interfere with the analysis. Feature extraction techniques, such as Count Vectorization and Term Frequency-Inverse Document Frequency (TF-IDF) transformation, convert textual data into numerical representations that can be used as input for machine learning models.

Balancing the dataset is another crucial step in sentiment analysis, especially when dealing with imbalanced data where one sentiment class significantly outnumbers the other. Techniques like upsampling the minority class help to create a balanced dataset, ensuring that the machine learning model does not become biased towards the majority class.

Evaluating the performance of the sentiment analysis model involves using metrics such as the F1 score, which provides a balance between precision and recall, and confusion matrices, which illustrate the model's performance in correctly classifying the sentiment classes. Additionally, visual tools like word clouds and ROC curves can offer deeper insights into the data distribution and model performance.

3 Methods

3.1 Data Collection and Preprocessing

The dataset for this study consists of labeled tweets, divided into training and test sets. Each tweet is annotated with a sentiment label indicating positive (0) or negative sentiment (1). We first clean the data by converting text to lowercase, removing special characters, URLs, and user mentions, and tokenizing the text into individual words.

To handle class imbalance, as seen in Figure 1, we apply the resampling technique. Specifically, we use upsampling to increase the number of minority class samples to match the majority class, creating a balanced dataset for training. This balanced dataset (Figure 2) ensures that the machine learning model does not favor the majority class.

3.2 Feature Extraction

Post preprocessing, we extract features using Count Vectorization and Term Frequency-Inverse Document Frequency (TF-IDF) transformation. Count Vectorization converts text into a matrix of token counts, while TF-IDF scales these counts by word frequency, highlighting significant yet less frequent words, as seen in Figures 3 and 4. These features provide a numerical representation of the text data, which is necessary for model training.

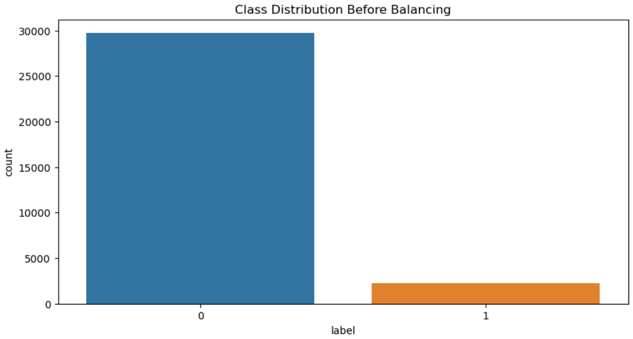

Figure 1. Class distribution before balancing

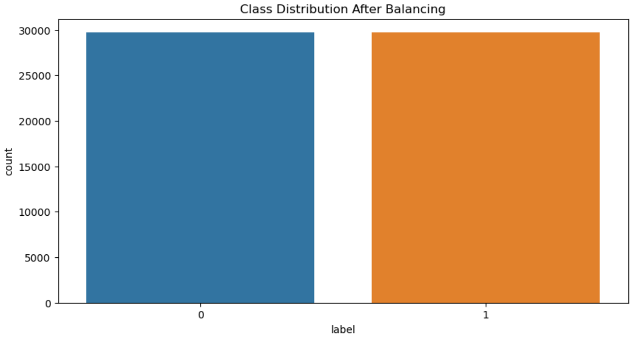

Figure 2. Class distribution after balancing

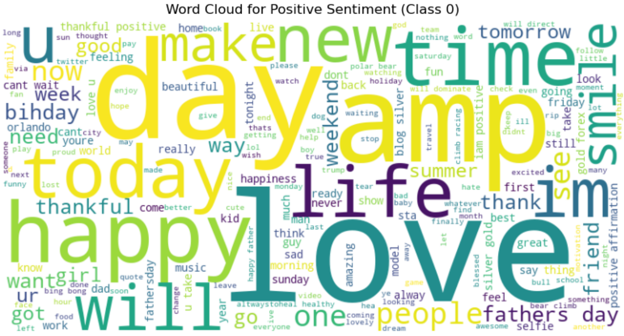

Figure 3. Word cloud for positive sentiment

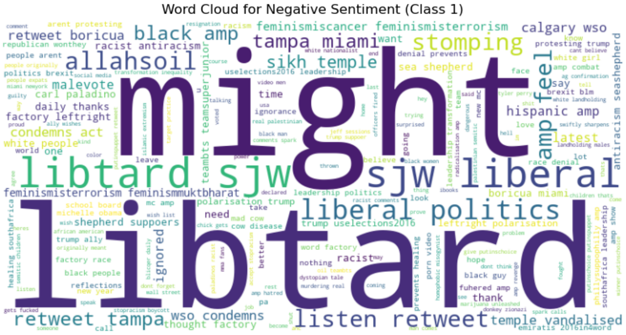

Figure 4. Word cloud for negative sentiment

3.3 *Model Training*

For sentiment classification, we employ logistic regression, a binary classification algorithm that models the probability of the positive class as a logistic function of input features. We construct a machine learning pipeline integrating feature extraction methods with the logistic regression classifier, ensuring consistent application during training and prediction.

3.4 *Model Evaluation*

We split the dataset into training and test sets with an 80-20 split. The model is trained on the training set and used to predict sentiment labels on the test set. We use the F1 score, balancing precision and recall, as our primary evaluation metric, and generate a confusion matrix to visualize the model's classification performance.

4 Results

After applying the above methods, we evaluated our logistic regression model on the test set. The final results of our sentiment analysis are summarized as follows:

The logistic regression model achieved an F1 score of 0.974974974974975, which indicates the balance between precision and recall for the sentiment classification task. This score demonstrates the effectiveness of our model in correctly identifying both positive and negative sentiments.

The confusion matrix (Figure 5) provides a detailed breakdown of the model's performance, showing the counts of true positives, true negatives, false positives, and false negatives. This matrix is essential for understanding the types of errors the model makes.

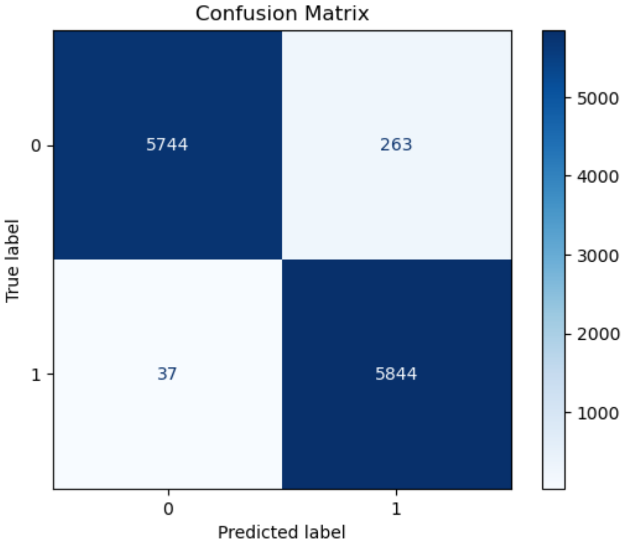

Figure 5. Confusion matrix

Finally, the ROC curve (Figure 6) illustrates the model's ability to distinguish between positive and negative sentiments across different classification thresholds. The area under the ROC curve was 0.9943688946243553, reflecting the model's overall performance.

These results confirm that our logistic regression model, com-

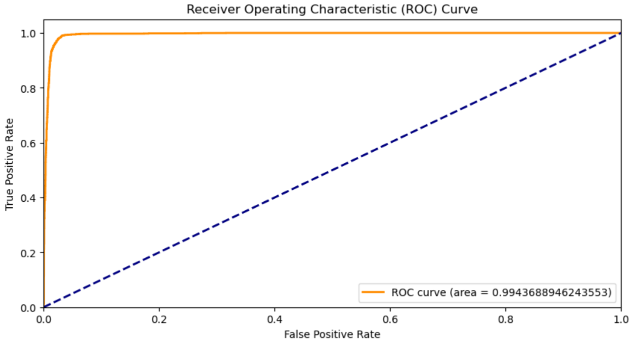

Figure 6. ROC curve

bined with robust data preprocessing and feature extraction techniques, is effective in performing sentiment analysis on tweets.

5 Discussion

The choice of using a Logistic Regression model for our sentiment analysis project over other classification models, such as Decision Trees, Artificial Neural Networks (ANNs), or k-Nearest Neighbors (k-NN), was driven by several considerations, including time complexity, space complexity, and the nature of our dataset.

Logistic Regression is a linear model that is particularly well-suited for binary classification tasks, like sentiment analysis, where the goal is to predict one of two possible outcomes (positive or negative sentiment). One of the key advantages of Logistic Regression is its simplicity and efficiency. The time complexity of Logistic Regression is $O(n \cdot m)$, where n is the number of features and m is the number of training samples. This linear time complexity makes Logistic Regression relatively fast, even for large datasets, and thus, it was a practical choice for our project, where computational resources and time were considerations.

In contrast, Decision Trees have a time complexity of $O(n \cdot m \cdot \log(m))$, which can become quite substantial as the dataset grows. Moreover, Decision Trees can be prone to overfitting, especially with noisy data, which can be a concern in natural language processing tasks. While techniques like pruning can mitigate this, they add to the computational overhead.

Artificial Neural Networks, although powerful, come with a significantly higher computational cost. The training time complexity of ANNs is typically $O(n \cdot m \cdot k)$, where k represents the number of neurons and layers. This can become computationally expensive and time-consuming, particularly without access to specialized hardware like GPUs. Furthermore, ANNs require extensive tuning of hyperparameters, such as learning rates and the number of layers, to perform optimally. This added complexity was not justified given the nature and size of our dataset.

k-Nearest Neighbors (k-NN) is another alternative, but it also has its drawbacks. The time complexity for training k-NN is $O(1)$ since it is a lazy learner, but the prediction time complexity is $O(n \cdot m)$. For large datasets, this makes k-NN impractical as each prediction requires computing distances to all training samples. Additionally, k-NN has high space complexity because it needs to store all training data, making it inefficient in terms of memory usage.

In comparison, Logistic Regression offers a good balance between simplicity, efficiency, and performance. It requires fewer computational resources and provides robust results with less risk of overfitting when regularized appropriately. Its probabilistic framework also provides insights into the confidence of the predictions, which is valuable for understanding the model's performance.

Considering these factors, Logistic Regression was deemed the most appropriate model for our sentiment analysis project. Its lower time and space complexities, combined with its effectiveness in binary classification tasks, aligned well with our project requirements and constraints.

6 Conclusion

In this project, we successfully implemented a sentiment analysis model using Logistic Regression to classify the sentiment of tweets as either positive or negative. The process involved several critical steps, including data cleaning, text preprocessing, feature extraction, and model training and evaluation. Each of these steps was meticulously carried out to ensure the robustness and reliability of our model.

Our model achieved a commendable F1 score, indicating its ability to balance precision and recall effectively. The confusion matrix and ROC curve further validated the model's performance, providing insights into its prediction accuracy and overall effectiveness. The area under the ROC curve (AUC) demonstrated the model's capability to distinguish between positive and negative sentiments accurately.

In conclusion, this project showcases the effective use of Logistic Regression for sentiment analysis, balancing simplicity and performance. The detailed steps and considerations outlined in this report provide a comprehensive understanding of the methodologies employed and the rationale behind the choices made. This project not only demonstrates the applicability of Logistic Regression in natural language processing tasks but also serves as a foundation for future work in this domain.